

AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1-10. (Canceled)

11. (Previously presented) A software, encoded on a computer readable medium, for generation of a computer code of at least one part of a computer application, in which the software generates the said computer code from a description of said at least one part of the computer application by distributing said description between several code generators according to modifiable distribution rules, each code generator translating the part of said description that it is provided with, in order to provide at least one part of the said computer code in a respective language.

12. (Previously presented) The software according to claim 11, splitting up said description in object classes, the software distributing said object classes between the code generators according to said distribution rules, each code generator translating the object classes that it is provided with, into said corresponding part of the said computer code.

13. (Previously presented) The software according to claim 12, moreover splitting up said description in dependencies between said object classes, and in the case of a dependency between two object classes each translated by a different code generator, the software makes antes said dependency be handled by two adapters that each translate it into a computer code for interfacing the computer codes produced by said code generators for said two object classes.

14. (Previously presented) The software according to claim 13, in which each of the two adapters produce said respective interfacing computer cede for a respective object class among said two object classes.

15. (Previously presented) The software according to claim 14, in which each of the two adapters inserts the respective interfacing computer code into the computer code produced by one of said code generators for said object class for which the adapter has produced said interfacing computer code.

16. (Previously presented) The software according to claim 15, in which said two adapters having to handle the dependency are chosen by the software following assignment rules associating, for the orientation of the dependency of said two object classes, an adapter corresponding to each of the code generators translating said two object classes, the said assignment rules being modifiable.

17. (Previously presented) The software according to claim 13, in which said two adapters having to handle the dependency are chosen by the software following assignment rules associating, for the orientation of the dependency of said two object classes, an adapter corresponding to each of the code generators translating said two object classes, said assignment rules being modifiable.

18. (Previously presented) The software according to claim 14, in which said two adapters having to handle the dependency are chosen by the software following assignment rules associating, for the orientation of the dependency of said two object classes, an adapter corresponding to each of the code generators translating said two object classes, said assignment rules being modifiable.

19. (Previously presented) The software according to claim 11, generating said computer code from said description made in a language organized in object classes, in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application, said services being not definable by said language, and the other classes of said language cannot have access to any one of said services except through said first classes.

20. (Previously presented) The software according to the claim 19, distributing said description between the code generators according to distribution rules associating at least some of said first classes or of said other classes of said language with at least one of said code generators.

21. (Previously presented) The software according to claim 20, splitting up said description in object classes, the software distributing said object classes between the code generators according to said distribution rules, each code generator translating the objects classes that it is provided with, into said corresponding part of said computer code and wherein the software splits up said description into first classes or into other classes of said language.

22. (Previously presented) The software according to the claim 17, generating said computer code from said description made in a language organized in object classes, in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application, said services being not definable by said language, and the other classes of said language cannot have access to any one of said services except through said first classes and wherein the software splits up said description in dependencies between said object classes from dependencies between said first classes or other classes of said language.

23. (Previously presented) The software according to the claim 17, generating said computer code from said description made in a language organized in object classes, in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application, said services being not definable by said language, and the other classes of said language cannot have access to any one of said services except through said first classes,

wherein the software distributes said description between the code generators according to distribution rules associating at least some of said first classes or of said

other classes of said language with at least one of said code generators, and

wherein the software splits up said description in dependencies between said object classes from dependencies between said first classes or other classes of said language.

24. (Previously presented) The software according to the claim 17, generating said computer code from said description made in a language organized in object classes, in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application, said services being not definable by said language, and the other classes of said language cannot have access to any one of said services except through said first classes,

wherein the software splits up said description into first classes or into other classes of said language,

wherein the software further splits up said description in dependencies between said object classes from dependencies between said first classes or other classes of said language, and

wherein the software distributes said description between the code generators according to distribution rules associating at least some of said first classes or of said other classes of said language with at least one of said code generators.

25. (Previously presented) A software description language, encoded on a computer readable medium, organized in classes enabling to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application, in which:

- the said services cannot be defined by said language, and
- the other classes cannot have access to any one of these technical or functional services except through said first classes.

26. (Previously presented) The software description language according to claim 25 of the type of an object-oriented language for computer application modeling.

27. (Previously presented) A software, encoded on a computer readable medium, enabling to graphically or textually build a computer application model and to provide a description of the model in a software description language organized in classes enabling to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application, in which:

- the said services cannot be defined by said language, and
- the other classes cannot have access to any one of these technical or functional services except through said first classes.

28. (Previously presented) The software according to claim 27, enabling to graphically or textually build a model of computer application human-machine interface.

29. (New) A software, encoded on either a random access memory (RAM) or a hard disk, for generation of a computer code of at least one part of a computer application, in which the software generates the said computer code from a description of said at least one part of the computer application by distributing said description between several code generators according to modifiable distribution rules, each code generator translating the part of said description that it is provided with, in order to provide at least one part of the said computer code in a respective language.

30. (New) A software description language, encoded on either a random access memory (RAM) or a hard disk, organized in classes enabling to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application, in which:

- the said services cannot be defined by said language, and

- the other classes cannot have access to any one of these technical or functional services except through said first classes.

31. (New) A software, encoded on either a random access memory (RAM) or a hard disk, enabling to graphically or textually build a computer application model and to provide a description of the model in a software description language organized in classes enabling to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application, in which:

- the said services cannot be defined by said language, and
- the other classes cannot have access to any one of these technical or functional services except through said first classes.